

# Café Orders & Inventory Tracker

## Goal:

Create a Python program to manage a small café's daily orders using **lists**, **dictionaries**, and **for loops only**.

## Given Data (Use this in your program):

```
menu = {  
    "tea": 120,  
    "coffee": 200,  
    "sandwich": 450,  
    "cookie": 150,  
    "cake_slice": 320  
}
```

```
inventory = {  
    "tea": 40,  
    "coffee": 35,  
    "sandwich": 20,  
    "cookie": 50,  
    "cake_slice": 25  
}
```

```
orders = [  
    {"tea": 2, "cookie": 1},  
    {"coffee": 1, "sandwich": 2},  
    {"tea": 3, "cake_slice": 1},  
    {"coffee": 2, "cookie": 2, "sandwich": 1},  
    {"tea": 1},  
    {"coffee": 1, "cake_slice": 2}  
]
```

## Tasks:

### 1. Validate Orders

- Go through each order.
- If an item is **not** in the menu → **ignore it**.
- If asked quantity is **greater than available inventory** → reduce to available quantity.
- Save the cleaned orders in a new list named **processed\_orders**.

### 2. Calculate Bill for Each Order

- For each processed order:
  - Multiply quantity × price.
- Store each total in a list called **order\_totals**.

### 3. Update Inventory

- Subtract sold items from the **inventory** as orders are processed.

### 4. Day Summary

Calculate and display:

- **Total revenue** (sum of order\_totals)
- **Items sold** (dictionary of item → total quantity)
- **Top selling item** (item with highest quantity sold)

### 5. Low Stock Report

- Find items where remaining inventory  $\leq 5$ .
- Store these items in a list named **low\_stock**.
- Print the list clearly.

### 6. Apply Simple Discount

- If an order total is **more than 1000 PKR**, apply **10% discount**.
- Create a new list: **order\_totals\_with\_discount**.
- Recalculate **total revenue with discount**.

## 7. Loyalty Points

- For each order, give **1 point per 100 PKR spent** (after discount).
- Save points for each order in **order\_points**.
- Also print **total points given** today.